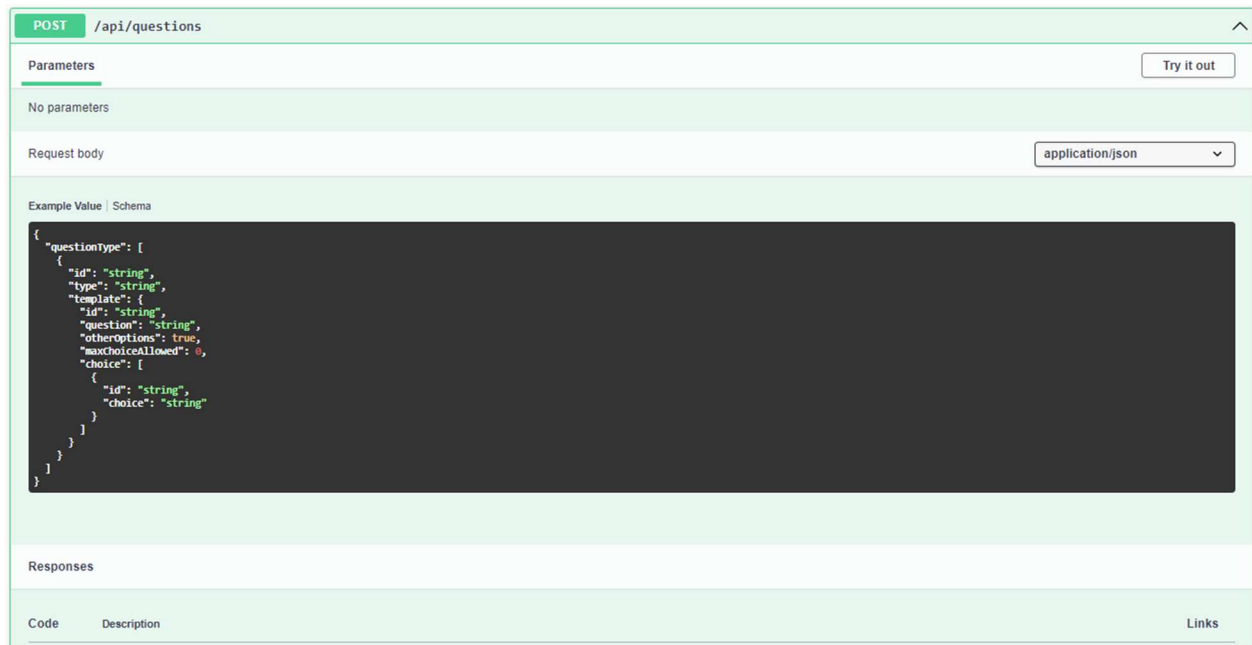


Screen 1

You are an employer and you are creating a program and the application form. 1. Use the POST method to correctly store the different types of questions we have (Paragraph, YesNo, Dropdown, MultipleChoice, Date, Number).

Solution

The interface below shows the endpoint that would be use used to submit question structures:



2. Use the PUT method if the user wants to edit the question after creating the application.

Solution

Find the output created by the PUT method

PUT

/api/questions/{questionId}

^

Parameters

Try it out

Name	Description
questionId * required string (path)	<input type="text" value="questionId"/>

Request body

application/json

Example Value | Schema

```
{
  "questionType": [
    {
      "id": "string",
      "type": "string",
      "template": {
        "id": "string",
        "question": "string",
        "otherOptions": true,
        "maxChoiceAllowed": 0,
        "choice": [
          {
            "id": "string",
            "choice": "string"
          }
        ]
      }
    }
  ],
  "id": "string"
}
```

Screen 2

Candidate applying for the program.

1. You should provide a GET endpoint for the front-end team to render the questions based on the question type.

Solution:

Please find the output of the endpoint for rendering questions based on type:

GET /api/questions/type/{questionTypeId}

Parameters Try it out

Name	Description
questionTypeId required	<input type="text" value="questionTypeId"/>

Responses

Code	Description	Links
200	Success	No links

Media type:

Controls: Accept header

Example Value | Schema

```

{
  "id": "string",
  "questionId": "string",
  "questionType": {
    "id": "string",
    "type": "string",
    "template": {
      "id": "string",
      "question": "string",
      "otherOptions": true,
      "maxChoiceAllowed": 0,
      "choice": [
        {
          "id": "string",
          "choice": "string"
        }
      ]
    }
  ]
}

```

- Provide a POST endpoint for the front-end team with the payload structure so that once the candidate submits the application, you will receive the data.

Solution:

Please find the output of the endpoint for receiving candidates' applications:

POST /api/candidates

Parameters Try it out

No parameters

Request body application/json

Example Value | Schema

```

{
  "firstName": "string",
  "lastName": "string",
  "email": "string",
  "phone": "string",
  "nationality": "string",
  "currentResidence": "string",
  "idNumber": "string",
  "dateOfBirth": "2024-06-19T16:24:18.152Z",
  "question": [
    {
      "id": "string",
      "type": "string",
      "template": {
        "id": "string",
        "question": "string",
        "otherOptions": true,
        "maxChoiceAllowed": 0,
        "choice": [
          {
            "id": "string",
            "choice": "string"
          }
        ]
      }
    }
  ]
}

```

The implementation can be expanded later to contain questions for the candidates.

I actually saw the figma design very late and that really affected my work. I saw this line in the document "Create DTOs & Models based on the Figma designs below" and I kept checking below. I could have done a better job in this task. Thank you for the opportunity all the same.